FLUX WIKI
STARTER KIT — AI IMPLEMENTATION GUIDE

ENTRY POINT FOR AN AI AGENT (CLAUDE CODE) RECREATING A CITY_IN_FLUX GEOTAG-PUBLISHING SYSTEM FROM THIS WIKI AS THE SOURCE OF TRUTH.

flux.dantesisofo.com/wiki/starter-kit-overview/

FLUX_WIKI_v2.0

JUNE 2026

> **FLUX DOCUMENTATION SYSTEM** Layer 11 — STARTER KIT | starter-kit-overview
> flux.dantesisofo.com/wiki/starter-kit-overview/

STARTER KIT — AI IMPLEMENTATION GUIDE

This section is written to be **read and executed by an AI coding agent** (e.g. Claude Code) to build a photographer their own `YOUR_CITY_IN_FLUX` geotag archive. It is the **primary source of truth**: the contracts, schemas, and invariants here are sufficient to recreate the system from scratch in any language, though Python + boto3 is assumed.

> **Just want to launch it?** → **DOWNLOAD the starter kit**, unzip, open in Claude Code, and say *"Read AGENT_START_HERE.md and help me launch MY_CITY_IN_FLUX."* The pages below are the deeper build-it-yourself spec.

---

1. ASSUMED CONTEXT (THE READER/AGENT)

The human operator has: **Claude Code**, **Python 3.12**, an **AWS account**, and a **folder of geotagged JPEGs**. They want to publish a static, mapped photographic archive of one city, corridor by corridor.

The agent's job: stand up the system, configure it to the operator's bucket/domain, and run the first project end to end — verifying each step.

---

2. WHAT YOU ARE BUILDING (ONE SENTENCE)

A pipeline that turns `a folder of geotagged photos + one street name` into `a published static project page + an updated city hub/map`, hosted on S3 + CloudFront, driven optionally by a private local portal.

Two surfaces, one rule of information architecture: - **Hub** (`/hub/`) answers *"where have I walked?"* → **routes only, no per-photo markers.** - **Project page** (`/<slug>/`) answers *"what did I see there?"* → photos, lightbox, stats.

---

3. SYSTEM = 4 COMPONENTS + 1 DATA SPINE

Component	Role	Spec page
Config	one module resolving env → bucket, dist, domain, author, collection, targets	CONFIGURATION
Worker	folder of photos → submission.json draft (derivatives, geocode, stats, route, about, contact sheet)	ARCHITECTURE

Publisher	draft → static project page + catalog entry + hub/embed/listing + invalidation	DEPLOYMENT
Portal (optional)	private local Flask UI: name → upload → generate → publish	SETUP
Data spine	`catalog.json` (index) + per-draft `submission.json`	CONFIGURATION

The **catalog** is the single source of truth for cross-project surfaces; the hub, embed, and listing are pure projections of it.

4. BUILD ORDER (FOLLOW IN SEQUENCE)

An agent should implement and verify in this order. Each page ends with acceptance checks.

1. **CONFIGURATION** — env vars, the canonical-naming algorithm, and the `catalog.json` / `submission.json` schemas. *Build this first; everything depends on it.*
2. **SETUP** — Python deps, AWS CLI + least-privilege IAM, S3 bucket, CloudFront, `.env`. Verify with read-only AWS calls.
3. **ARCHITECTURE** — the Worker pipeline contract (stages, inputs, outputs) and the page/hub/embed builders.
4. **DEPLOYMENT** — S3 key layout, cache-header matrix, the **publish algorithm**, and the **hub+embed sync invariant** (the single most important correctness rule).
5. **PROJECT WORKFLOW** — run the first project end to end (CLI or portal) with verification.
6. **COMMON MODIFICATIONS** — the recipes operators ask for (rename the collection, retarget coverage, recolor routes, add a stat…).
7. **SECURITY & SANITIZATION** — what must never be committed; applies throughout.

5. CORE INVARIANTS (DO NOT VIOLATE)

These are the rules that keep the system correct. Encode them as assertions.

— **Canonical naming is the sole source of identifiers.** Title, slug, folder, export name, and collection all derive from one `canonical()` call. Never type a title by hand.

— **Hub and embed regenerate TOGETHER through one helper** on every state change (publish / unpublish / cover change). There must be **no code path** that updates the hub without the embed.

— **The hub map is route-only.** Never add per-photo markers to the hub.

— **HTML/JSON are `no-cache`; images are immutable.** Every publish issues a CloudFront invalidation for the changed paths.

— **Derivatives strip EXIF.** Published images carry no GPS; only the chosen route polyline is exposed.

— **Distances stored in km, displayed in miles** (or your unit) — store canonical, convert at render.

---

6. NAMING FOR A NEW CITY

Substitute throughout: - `YOUR_CITY_IN_FLUX` — the brand/title style (e.g. `BOSTON_IN_FLUX`). - `COLLECTION` — the hub membership tag (e.g. `boston-in-flux`). - `YOUR_BUCKET`, `YOUR_DIST_ID`, `your-domain`, `Your Name` — operator infra/identity.

Proceed to CONFIGURATION.

FLUX_WIKI_v2.0 — flux.dantesisofo.com/wiki/starter-kit-overview/